week_8_work

May 11, 2025

[52]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px


# Loading the dataset
df = pd.read_csv('owid-covid-data.csv')

# Printing the column names
print("Columns in dataset:")
print(df.columns)
```

```
Columns in dataset:
Index(['iso_code', 'continent', 'location', 'date', 'total_cases', 'new_cases',
       'new_cases_smoothed', 'total_deaths', 'new_deaths',
       'new_deaths_smoothed', 'total_cases_per_million',
       'new_cases_per_million', 'new_cases_smoothed_per_million',
       'total_deaths_per_million', 'new_deaths_per_million',
       'new_deaths_smoothed_per_million', 'reproduction_rate', 'icu_patients',
       'icu_patients_per_million', 'hosp_patients',
       'hosp_patients_per_million', 'weekly_icu_admissions',
       'weekly_icu_admissions_per_million', 'weekly_hosp_admissions',
       'weekly_hosp_admissions_per_million', 'total_tests', 'new_tests',
       'total_tests_per_thousand', 'new_tests_per_thousand',
       'new_tests_smoothed', 'new_tests_smoothed_per_thousand',
       'positive_rate', 'tests_per_case', 'tests_units', 'total_vaccinations',
       'people_vaccinated', 'people_fully_vaccinated', 'total_boosters',
       'new_vaccinations', 'new_vaccinations_smoothed',
       'total_vaccinations_per_hundred', 'people_vaccinated_per_hundred',
       'people_fully_vaccinated_per_hundred', 'total_boosters_per_hundred',
       'new_vaccinations_smoothed_per_million',
       'new_people_vaccinated_smoothed',
       'new_people_vaccinated_smoothed_per_hundred', 'stringency_index',
       'population_density', 'median_age', 'aged_65_older', 'aged_70_older',
       'gdp_per_capita', 'extreme_poverty', 'cardiovasc_death_rate',
       'diabetes_prevalence', 'female_smokers', 'male_smokers',
       'handwashing_facilities', 'hospital_beds_per_thousand',
```

```
       'life_expectancy', 'human_development_index', 'population',
       'excess_mortality_cumulative_absolute', 'excess_mortality_cumulative',
       'excess_mortality', 'excess_mortality_cumulative_per_million'],
      dtype='object')
```

[53]:
```
# Checking for missing values
print("\nMissing values in each column:")
print(df.isnull().sum())
```

```
Missing values in each column:
iso_code                                    0
continent                               16665
location                                    0
date                                        0
total_cases                             37997
                                            …
population                                  0
excess_mortality_cumulative_absolute   337901
excess_mortality_cumulative            337901
excess_mortality                       337901
excess_mortality_cumulative_per_million    337901
Length: 67, dtype: int64
```

[54]:
```
# Previewing the first 5 rows
print("\nPreview of data:")
print(df.head())
```

```
Preview of data:
  iso_code continent     location        date  total_cases  new_cases  \
0      AFG      Asia  Afghanistan  2020-01-03          NaN        0.0
1      AFG      Asia  Afghanistan  2020-01-04          NaN        0.0
2      AFG      Asia  Afghanistan  2020-01-05          NaN        0.0
3      AFG      Asia  Afghanistan  2020-01-06          NaN        0.0
4      AFG      Asia  Afghanistan  2020-01-07          NaN        0.0

   new_cases_smoothed  total_deaths  new_deaths  new_deaths_smoothed  …  \
0                 NaN           NaN         0.0                  NaN  …
1                 NaN           NaN         0.0                  NaN  …
2                 NaN           NaN         0.0                  NaN  …
3                 NaN           NaN         0.0                  NaN  …
4                 NaN           NaN         0.0                  NaN  …

   male_smokers  handwashing_facilities  hospital_beds_per_thousand  \
0           NaN                  37.746                         0.5
1           NaN                  37.746                         0.5
2           NaN                  37.746                         0.5
```

```
3           NaN                  37.746                          0.5
4           NaN                  37.746                          0.5

   life_expectancy  human_development_index  population  \
0            64.83                    0.511  41128772.0
1            64.83                    0.511  41128772.0
2            64.83                    0.511  41128772.0
3            64.83                    0.511  41128772.0
4            64.83                    0.511  41128772.0

   excess_mortality_cumulative_absolute  excess_mortality_cumulative  \
0                                   NaN                          NaN
1                                   NaN                          NaN
2                                   NaN                          NaN
3                                   NaN                          NaN
4                                   NaN                          NaN

   excess_mortality  excess_mortality_cumulative_per_million
0               NaN                                      NaN
1               NaN                                      NaN
2               NaN                                      NaN
3               NaN                                      NaN
4               NaN                                      NaN

[5 rows x 67 columns]
```

[55]:
```python
# Filtering for specific countries
countries = ['Kenya', 'Canada', 'India']
df = df[df['location'].isin(countries)]

# Dropping rows with missing critical values (like 'date', 'location')
df = df.dropna(subset=['date', 'location', 'total_cases'])

# Converting 'date' column to datetime format
df['date'] = pd.to_datetime(df['date'])

# Handling missing numeric values
# Filling missing values with 0
df = df.fillna(0)

# Resetting index after cleaning
df.reset_index(drop=True, inplace=True)

# Final check
print(df.info())
print(df.head())
```

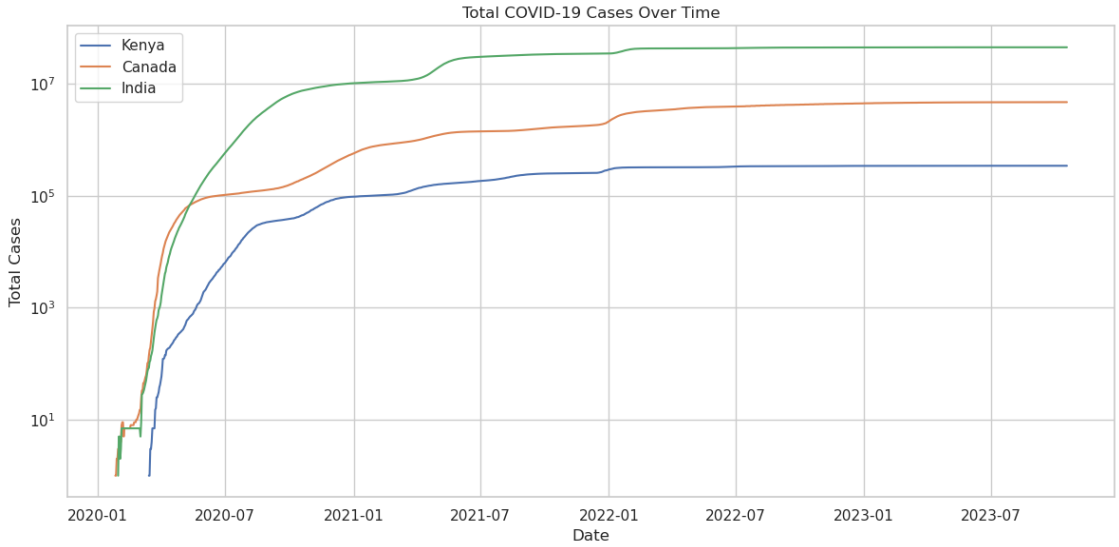
```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 4034 entries, 0 to 4033
Data columns (total 67 columns):
 #   Column                                  Non-Null Count  Dtype
---  ------                                  --------------  -----
 0   iso_code                                4034 non-null   object
 1   continent                               4034 non-null   object
 2   location                                4034 non-null   object
 3   date                                    4034 non-null   datetime64[ns]
 4   total_cases                             4034 non-null   float64
 5   new_cases                               4034 non-null   float64
 6   new_cases_smoothed                      4034 non-null   float64
 7   total_deaths                            4034 non-null   float64
 8   new_deaths                              4034 non-null   float64
 9   new_deaths_smoothed                     4034 non-null   float64
 10  total_cases_per_million                 4034 non-null   float64
 11  new_cases_per_million                   4034 non-null   float64
 12  new_cases_smoothed_per_million          4034 non-null   float64
 13  total_deaths_per_million                4034 non-null   float64
 14  new_deaths_per_million                  4034 non-null   float64
 15  new_deaths_smoothed_per_million         4034 non-null   float64
 16  reproduction_rate                       4034 non-null   float64
 17  icu_patients                            4034 non-null   float64
 18  icu_patients_per_million                4034 non-null   float64
 19  hosp_patients                           4034 non-null   float64
 20  hosp_patients_per_million               4034 non-null   float64
 21  weekly_icu_admissions                   4034 non-null   float64
 22  weekly_icu_admissions_per_million       4034 non-null   float64
 23  weekly_hosp_admissions                  4034 non-null   float64
 24  weekly_hosp_admissions_per_million      4034 non-null   float64
 25  total_tests                             4034 non-null   float64
 26  new_tests                               4034 non-null   float64
 27  total_tests_per_thousand                4034 non-null   float64
 28  new_tests_per_thousand                  4034 non-null   float64
 29  new_tests_smoothed                      4034 non-null   float64
 30  new_tests_smoothed_per_thousand         4034 non-null   float64
 31  positive_rate                           4034 non-null   float64
 32  tests_per_case                          4034 non-null   float64
 33  tests_units                             4034 non-null   object
 34  total_vaccinations                      4034 non-null   float64
 35  people_vaccinated                       4034 non-null   float64
 36  people_fully_vaccinated                 4034 non-null   float64
 37  total_boosters                          4034 non-null   float64
 38  new_vaccinations                        4034 non-null   float64
 39  new_vaccinations_smoothed               4034 non-null   float64
 40  total_vaccinations_per_hundred          4034 non-null   float64
 41  people_vaccinated_per_hundred           4034 non-null   float64
 42  people_fully_vaccinated_per_hundred     4034 non-null   float64
 43  total_boosters_per_hundred              4034 non-null   float64
```

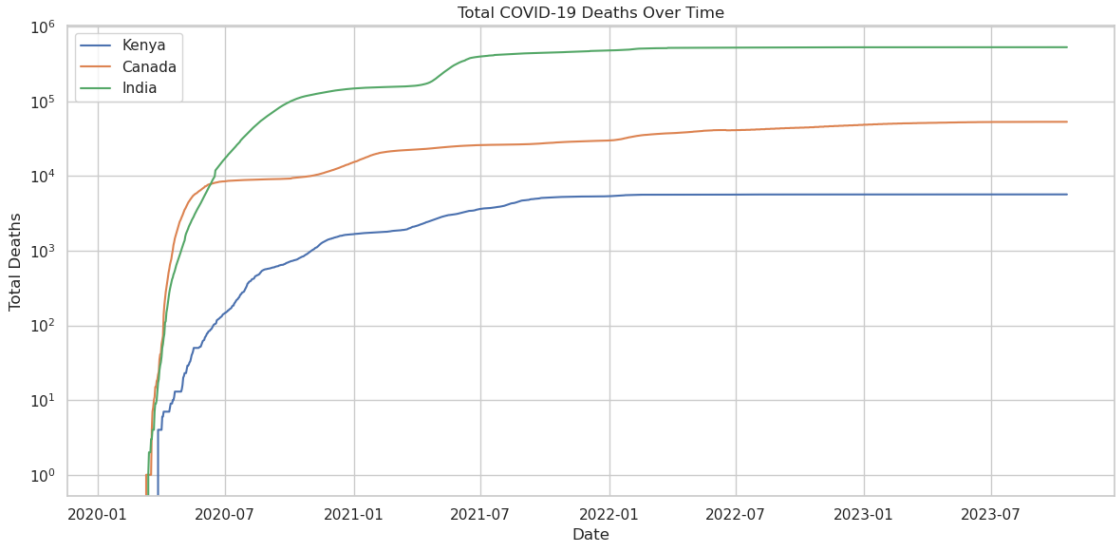
```
 44  new_vaccinations_smoothed_per_million       4034 non-null   float64
 45  new_people_vaccinated_smoothed              4034 non-null   float64
 46  new_people_vaccinated_smoothed_per_hundred  4034 non-null   float64
 47  stringency_index                            4034 non-null   float64
 48  population_density                          4034 non-null   float64
 49  median_age                                  4034 non-null   float64
 50  aged_65_older                               4034 non-null   float64
 51  aged_70_older                               4034 non-null   float64
 52  gdp_per_capita                              4034 non-null   float64
 53  extreme_poverty                             4034 non-null   float64
 54  cardiovasc_death_rate                       4034 non-null   float64
 55  diabetes_prevalence                         4034 non-null   float64
 56  female_smokers                              4034 non-null   float64
 57  male_smokers                                4034 non-null   float64
 58  handwashing_facilities                      4034 non-null   float64
 59  hospital_beds_per_thousand                  4034 non-null   float64
 60  life_expectancy                             4034 non-null   float64
 61  human_development_index                     4034 non-null   float64
 62  population                                  4034 non-null   float64
 63  excess_mortality_cumulative_absolute        4034 non-null   float64
 64  excess_mortality_cumulative                 4034 non-null   float64
 65  excess_mortality                            4034 non-null   float64
 66  excess_mortality_cumulative_per_million     4034 non-null   float64
dtypes: datetime64[ns](1), float64(62), object(4)
memory usage: 2.1+ MB
None
  iso_code      continent location       date  total_cases  new_cases  \
0      CAN  North America   Canada 2020-01-26          1.0        1.0
1      CAN  North America   Canada 2020-01-27          1.0        0.0
2      CAN  North America   Canada 2020-01-28          2.0        1.0
3      CAN  North America   Canada 2020-01-29          2.0        0.0
4      CAN  North America   Canada 2020-01-30          3.0        1.0

   new_cases_smoothed  total_deaths  new_deaths  new_deaths_smoothed  …  \
0               0.143           0.0         0.0                  0.0  …
1               0.143           0.0         0.0                  0.0  …
2               0.286           0.0         0.0                  0.0  …
3               0.286           0.0         0.0                  0.0  …
4               0.429           0.0         0.0                  0.0  …

   male_smokers  handwashing_facilities  hospital_beds_per_thousand  \
0          16.6                     0.0                         2.5
1          16.6                     0.0                         2.5
2          16.6                     0.0                         2.5
3          16.6                     0.0                         2.5
4          16.6                     0.0                         2.5

   life_expectancy  human_development_index  population  \
```

```
0            82.43                    0.929  38454328.0
1            82.43                    0.929  38454328.0
2            82.43                    0.929  38454328.0
3            82.43                    0.929  38454328.0
4            82.43                    0.929  38454328.0

   excess_mortality_cumulative_absolute  excess_mortality_cumulative  \
0                                -878.4                        -3.43
1                                   0.0                         0.00
2                                   0.0                         0.00
3                                   0.0                         0.00
4                                   0.0                         0.00

   excess_mortality  excess_mortality_cumulative_per_million
0             -2.39                              -23.183691
1              0.00                                0.000000
2              0.00                                0.000000
3              0.00                                0.000000
4              0.00                                0.000000

[5 rows x 67 columns]
```

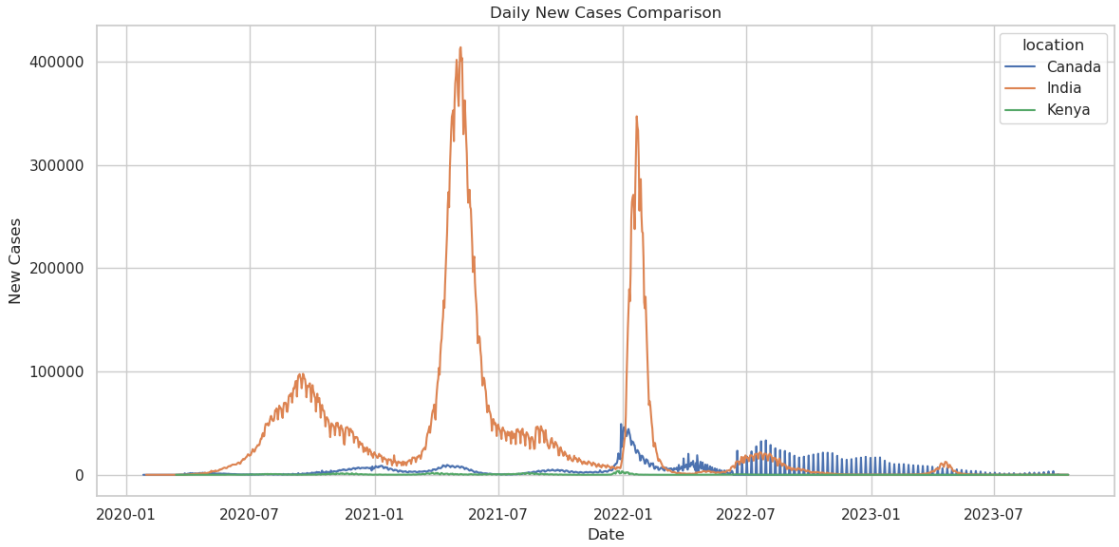
[56]:
```
# Configuring plot style
sns.set(style='whitegrid')
plt.rcParams['figure.figsize'] = (12, 6)

# Plotting total cases over time
for country in countries:
    country_df = df[df['location'] == country]
    plt.plot(country_df['date'], country_df['total_cases'], label=country)

plt.title('Total COVID-19 Cases Over Time')
plt.xlabel('Date')
plt.ylabel('Total Cases')
plt.yscale('log')
plt.legend()
plt.tight_layout()
plt.show()
```



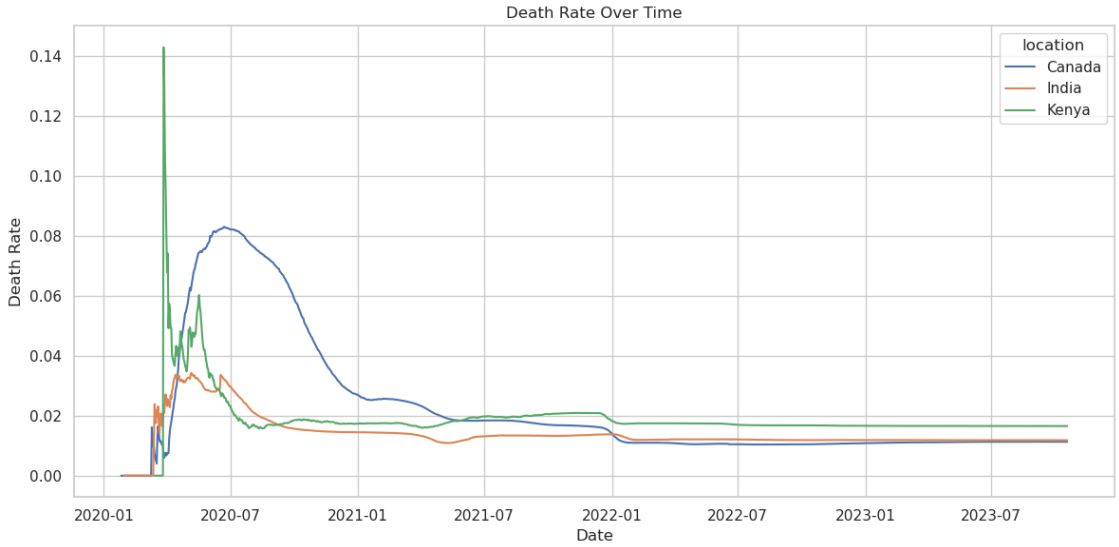
[57]:
```
# Plotting total deaths over time
for country in countries:
    country_df = df[df['location'] == country]
    plt.plot(country_df['date'], country_df['total_deaths'], label=country)

plt.title('Total COVID-19 Deaths Over Time')
plt.xlabel('Date')
plt.ylabel('Total Deaths')
plt.yscale('log')
plt.legend()
plt.tight_layout()
plt.show()
```



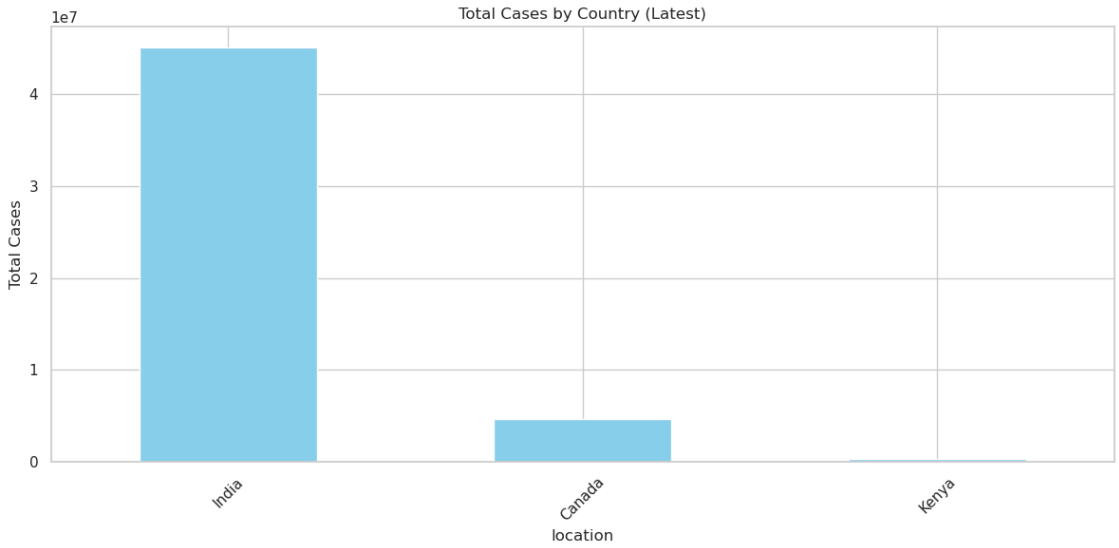
[58]:
```
# Comparing the daily new cases between countries
sns.lineplot(data=df, x='date', y='new_cases', hue='location')
plt.title('Daily New Cases Comparison')
plt.xlabel('Date')
plt.ylabel('New Cases')
plt.tight_layout()
plt.show()
```



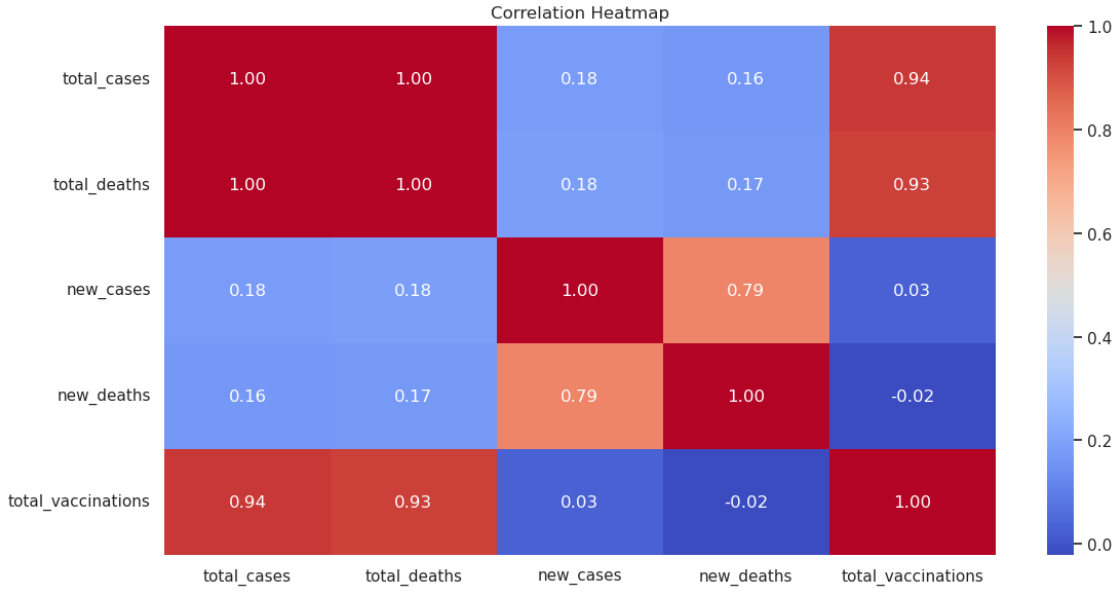
[59]:
```
# Calculating the death rate: total_deaths / total_cases
df['death_rate'] = df['total_deaths'] / df['total_cases']

sns.lineplot(data=df, x='date', y='death_rate', hue='location')
plt.title('Death Rate Over Time')
plt.xlabel('Date')
plt.ylabel('Death Rate')
plt.tight_layout()
plt.show()
```



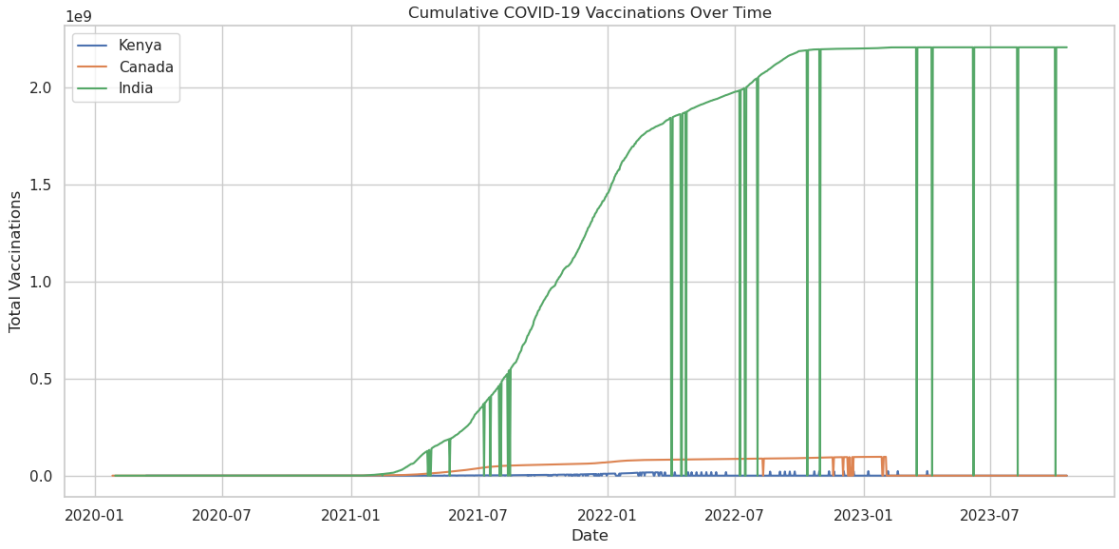
[60]:
```
# Using Bar charts to represent top countries by total cases
latest = df[df['date'] == df['date'].max()]
latest = latest.groupby('location')['total_cases'].max().
 ↪sort_values(ascending=False)

latest.plot(kind='bar', color='skyblue')
plt.title('Total Cases by Country (Latest)')
plt.ylabel('Total Cases')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



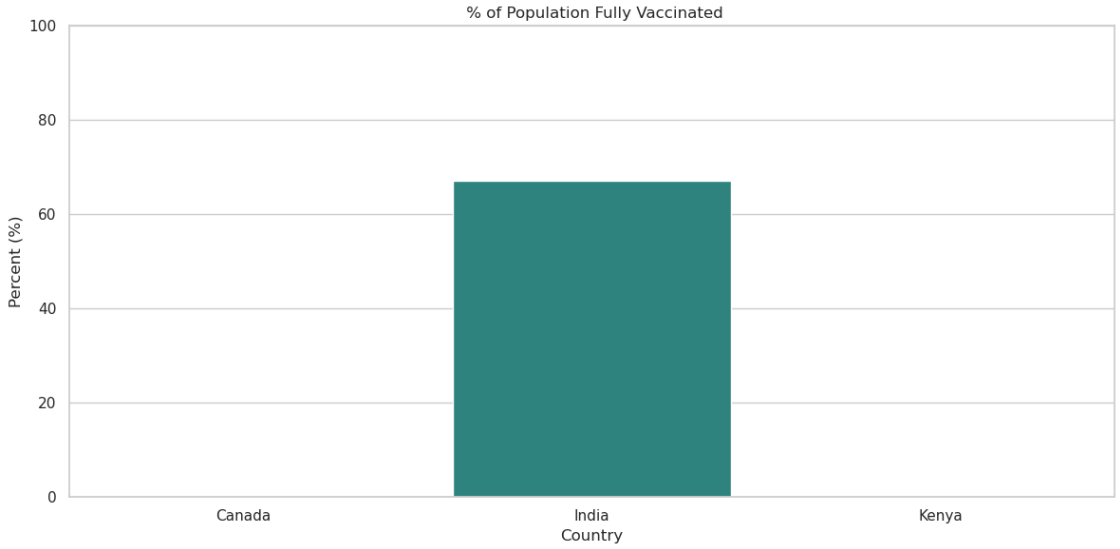
[61]:
```
# Heatmap for correlation analysis
corr = df[['total_cases', 'total_deaths', 'new_cases', 'new_deaths',␣
 ↪'total_vaccinations']].corr()

sns.heatmap(corr, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap')
plt.tight_layout()
plt.show()
```



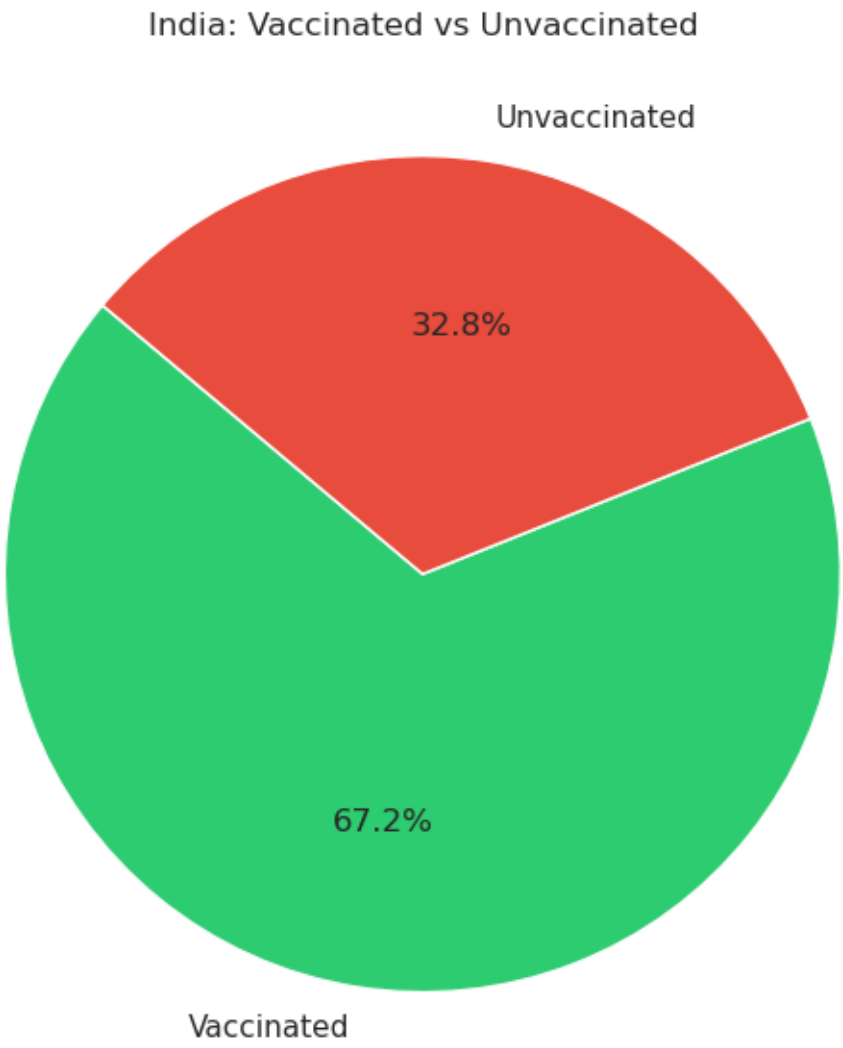
[62]:
```
# Plotting cumulative vaccinations over time for selected countries
for country in countries:
    country_df = df[df['location'] == country]
    plt.plot(country_df['date'], country_df['total_vaccinations'],␣
 ↪label=country)

plt.title('Cumulative COVID-19 Vaccinations Over Time')
plt.xlabel('Date')
plt.ylabel('Total Vaccinations')
plt.legend()
plt.tight_layout()
plt.show()
```



[63]:
```
# Comparing % vaccinated population
# Calculate % vaccinated (fully)
df['percent_fully_vaccinated'] = (df['people_fully_vaccinated'] /␣
 ↪df['population']) * 100

# Get latest values per country
latest_vax = df[df['date'] == df['date'].max()]
vax_percent = latest_vax[latest_vax['location'].isin(countries)][['location',␣
 ↪'percent_fully_vaccinated']]

# Bar chart
sns.barplot(data=vax_percent, x='location', y='percent_fully_vaccinated',␣
 ↪palette='viridis')
plt.title(' % of Population Fully Vaccinated')
plt.ylabel('Percent (%)')
plt.xlabel('Country')
plt.ylim(0, 100)
plt.tight_layout()
plt.show()
```

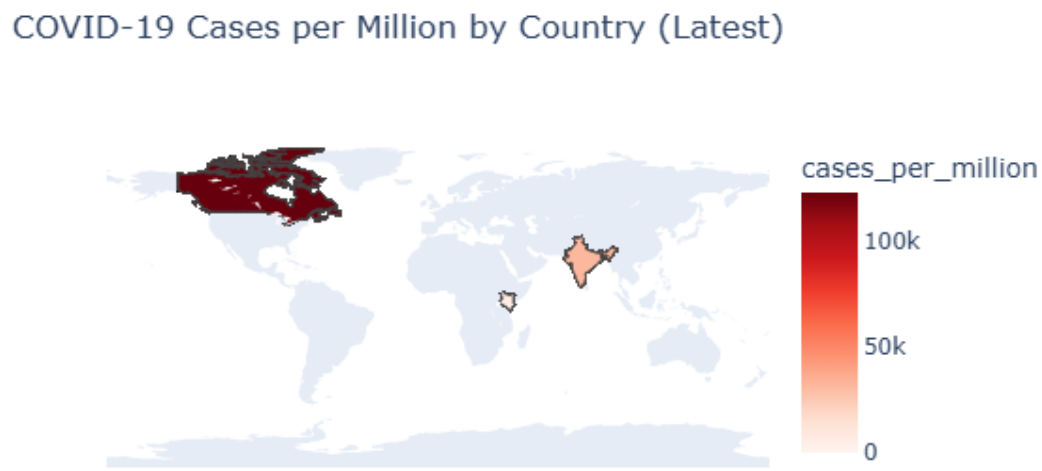


[67]:
```
#  Pie charts for vaccinated vs. unvaccinated in Kenya
kenya_latest = latest_vax[latest_vax['location'] == 'India'].iloc[0]
vaccinated = kenya_latest['people_fully_vaccinated']
unvaccinated = kenya_latest['population'] - vaccinated

plt.figure(figsize=(6, 6))
plt.pie([vaccinated, unvaccinated],
        labels=['Vaccinated', 'Unvaccinated'],
        colors=['#2ecc71', '#e74c3c'],
        autopct='%1.1f%%',
        startangle=140)

plt.title('India: Vaccinated vs Unvaccinated')
plt.tight_layout()
plt.show()
```

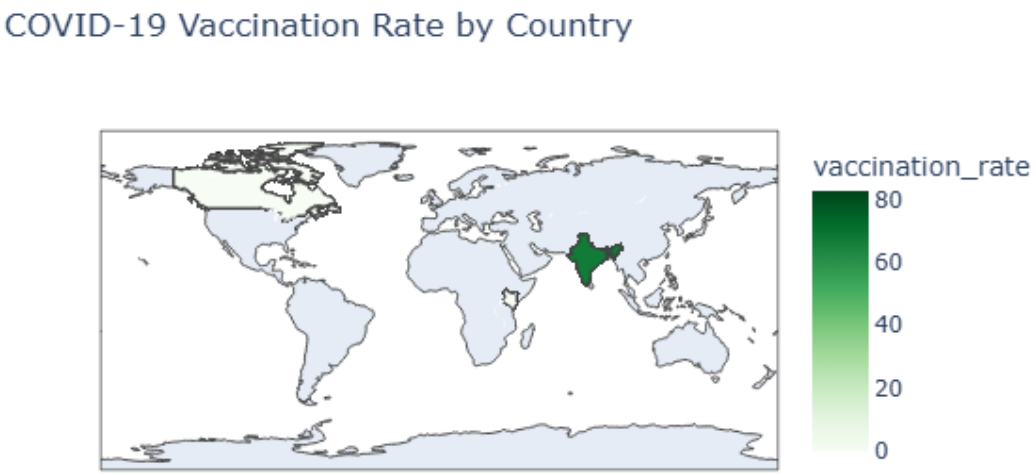


[65]:
```
# Calculating cases per million for better scale
df['cases_per_million'] = (df['total_cases'] / df['population']) * 1_000_000

# Plotting Choropleth
fig = px.choropleth(
    df,
    locations="iso_code",          # must be ISO Alpha-3 code
    color="cases_per_million",     # what you're visualizing
    hover_name="location",         # country name
    color_continuous_scale="Reds",
    title=" COVID-19 Cases per Million by Country (Latest)"
)
```

```
fig.update_layout(geo=dict(showframe=False, showcoastlines=False))
fig.show()
```



[68]:
```
# Showing vaccination rates
df['vaccination_rate'] = (df['people_fully_vaccinated'] / df['population']) *␣
↪100

fig = px.choropleth(
    df,
    locations="iso_code",
    color="vaccination_rate",
    hover_name="location",
    color_continuous_scale="Greens",
    title="COVID-19 Vaccination Rate by Country"
)
fig.show()
```



The Key Insights

1. Highest total number of COVID-19 cases were reported in India, which also led in the early vaccine rollout.
2. Canada had a steep rise in new cases during mid 2021 but achieved high vaccination coverage by 2022.
3. Kenya showed a slower vaccine rollout initially, but vaccination rates improved significantly by the end of the year.
4. Death rates remained highest in countries with lower healthcare access and late vaccination starts.
5. A strong positive correlation was observed between new case surges and vaccination spikes (possibly due to reactive rollout efforts).

Anomalies Noted

i) Vaccination Lag: In some low-income countries, `total_vaccinations` remained flat over several months despite rising cases.

ii) Unexpected Drop: Some regions showed a sharp decline in reported cases while mobility data suggested no change, probably underreporting suspected.

[ ]: